005306.P091 PATENT

UNITED STATES PATENT APPLICATION

for

METHOD AND SYSTEM TO FACILITATE DEVELOPMENT OF CUSTOMER-SPECIFIC BUSINESS PROCESS MODELS

Applicants:

Eric Berg
Krishna Kilambi

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard
Los Angeles, CA 90026-1026
(408) 720-8598

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EV336584443US
Date of Deposit June 23, 2003
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Michelle Begay
(Typed or printed name of person mailing paper or fee)
Michelle Begay
(Signature of person mailing paper or fee)

METHOD AND APPARATUS TO FACILITATE DEVELOPMENT OF A CUSTOMER-SPECIFIC BUSINESS PROCESS MODEL

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 60/392,443, filed June 27, 2002.

FIELD OF THE INVENTION

[0002] This invention relates generally to business process modeling, and more particularly to facilitating development of a customer-specific business process model.

COPYRIGHT NOTICE/PERMISSION

[0003] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2001, Siebel Systems, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

[0004] The Internet is transforming corporate architectures. Businesses are now recreating themselves to become more efficient and to find new business opportunities. For example, businesses are moving existing processes such as order entry processes onto the Internet while simultaneously incorporating entirely new processes such as email-based customer support into their computing systems. By shifting some tasks onto the Web and then integrating them with existing client/server or mainframe applications, businesses are transforming into an eBusiness with a unified enterprise computing system.

[0005] The development of a unified enterprise computing system is a long and complex process, especially for large businesses. Extensive planning is needed to determine the amount of configuration required to meet the organization's business strategy and to standardize processes across the organization. In addition, a typical organization has numerous business requirements, and it is a challenge to uncover these business requirements and implement them in a controlled phase approach.

[0006] Current providers of eBusiness applications have not been able to achieve much success in reducing cost and complexity of application integration and accelerating time to deployment. One of the reasons for this inability is that a company is rarely convinced that an eBusiness application offered by the provider satisfies the company's business needs. As a result, the provider has to go through a lengthy process of collecting the company's business requirements, making significant changes to the application in accordance to the business requirements and deploying the modified application. Furthermore, because different organizations have different business requirements and employ different strategy, the provider typically has to repeat this inefficient and costly process for each customer.

SUMMARY OF THE INVENTION

[0007] The present invention relates to various aspects for manipulating business process data and presenting the business process data to a user.

[0008] According to one aspect of the present invention, a business process model pertaining to an application product is associated with a set of views that illustrate the realization of the business process in the application product. Further, the business process model is displayed to the user together with the set of views to enhance the user's understanding of the business process provided by the application product.

[0009] The present invention describes systems, clients, servers, methods, and machine-readable media of varying scope. In addition to the aspects of the present invention described in this summary, further aspects of the invention will become apparent by reference to the drawings and by reading the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0011] **Figure 1** is a block diagram of one embodiment of a business process modeling system.

[0012] **Figure 2** is flow diagram of one embodiment of a process for facilitating the presentation of business process information.

[0013] **Figure 3** is a flow diagram of one embodiment of a process for associating a business process model with a set of views.

[0014] **Figure 4** illustrates a user interface for presenting business process data to a user, according to one embodiment of the present invention.

[0015] **Figures 5A – 5C** illustrate exemplary user interfaces facilitating an addition of a new view to a diagram, according to one embodiment of the present invention.

[0016] **Figure 6** is a flow diagram of a process for facilitating the navigation to a view in an application product.

[0017] **Figure 7** is a flow diagram of a process for facilitating the development of a customer-specific application product.

[0018] **Figure 8** is a block diagram of an exemplary computer system 800 that may be used to perform one or more of the operations described herein.

DETAILED DESCRIPTION OF THE INVENTION

[0019] In the following description, numerous details are set forth. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

[0020] Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0021] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0022] The present invention also relates to apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0023] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

[0024] A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory ("ROM"); random access memory ("RAM"); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

<u>Overview</u>

[0025] The present invention relates to various aspects for manipulating business process data and presenting the business process data to a user. In one embodiment, the business process data is first prepared for the presentation and then displayed to the user in a desired format. During the preparation phase, an application product is identified and a business process model pertaining to the application product is associated with a set of views that illustrate the realization of a business process in the application product. The business process model is created in a modeling language. A view corresponds to a specific user interface within the application product. In one embodiment, the view is an image representing a corresponding user interface. As will be discussed in greater detail below, the association of the business process model with the set of views is accomplished, in one embodiment, by creating a view for each user interface, storing an identifier of each view in a repository and creating in the repository a link between each view identifier and one or more steps (also referred to herein as activities) contained in the business process model.

[0026] During the presentation phase, the business process model is displayed to the user together with a set of views. The set of views may be selected from the views previously linked to the data process model. In one embodiment, the set of views presented to the user can be modified. That is, the user can request to delete some views, add new views or replace some views with different views. In one embodiment, in response to a user request to navigate to a specific view in the application product, a corresponding portion of the application program is executed and the user is presented with an actual user interface that the user can interact with (e.g., enter data, click buttons, etc.). As a result, the user can observe the realization of the business process in the application product and determine whether this business process satisfies the desired functionality.

[0027] In one embodiment, the user is provided with an option to request a transfer of business process data into a business requirements database. When the user submits a request for transfer, the business process data is transformed into a set of business requirements that are transferred to the business requirements database. In one embodiment, relationships between the business requirements are defined based on the structure of the business process model and maintained in the business requirements database. In one embodiment, when the business process model is modified, the business requirements stored in the business requirements database are modified accordingly. In another embodiment, the business requirements are modified directly in the business requirements database based on user input. The modified business requirements can then be exported to an external file that may subsequently be used to create a new business process model satisfying the modified business requirements.

[0028] In one embodiment, a development of a customer-specific business process model is facilitated using the presentation of business process information as described above. In this embodiment, the business process information pertains to a standard application product developed for a specific industry based on the best practices found in this industry. A business process model created for the industry standard application product is displayed to a customer with a set of views that illustrate the realization of a business process in the application product. This presentation of business process data enhances the customer's understanding of the functionality provided by the standard application product, assists in mapping of the customer's business requirements to standard functions and maximizes the use of standard functionality in the customized product.

[0029] In one embodiment, the development of a customer-specific application product is facilitated by transforming the business process model associated with the standard application product into a set of standard business requirements and storing the set of standard business requirements in the business requirements database. Further, when the standard business process model is modified according to the customer's needs, a set of customer-specific business requirements is created using the modified business process model and stored in the business requirements database. The two sets of business requirements can then be compared and analyzed to determine which components of the standard application product need to be modified to provide the functionality desired by the customer.

Business Process Modeling System

[0030] **Figure 1** is a block diagram of one embodiment of a business process modeling system 100. System 100 includes a modeling tool 104, a business requirements database 110, a linkage engine 106, and a linkage repository 108.

[0031] Modeling tool 104 is responsible for creating a business process model for an application product 102. Modeling tool 104 may be any known in the art modeling tool. Modeling tool 104 creates a business process model using a modeling language such as a unified modeling language (UML).

[0032] Business requirements database 110 stores various business requirements. Business requirements database 110 may be a part of a commercially available requirement management tool or an independent database such as a relational database, a flat-file database, a network database, or a hierarchical database. Data stored in business requirements database 110 may be received from modeling tool 104 and/or entered by the user.

[0033] Linkage engine 108 is responsible for preparing to the presentation of business process information to the user and for enabling the presentation of the business process information in such a manner as to enhance the user's understanding of a business process. During the preparation phase, linkage engine 108 associates a business process model created by modeling tool 104 with a group of views associated with application product 102. In one embodiment, the association between the business process model and the views is accomplished using linkage repository 108 that stores view identifiers (e.g., view names) and maintains a relationship between each view identifier and one or more activities contained in the business process model.

[0034] During the presentation phase, linkage engine 106 allows the user to select a set of views for display with the business process model. In addition, linkage engine 106 provides the user with an option to navigate to a specific view in the application product 102. That is, when the user requests navigation, linkage engine 106 sends a command to application product 102, triggering the display of an actual user interface in the execution mode.

[0035] Linkage engine 108 is further responsible for linking modeling tool 104 to business requirements database 110. In particular, linkage engine 108 allows the user to request transfer of business process data to business requirements database 100. Upon receiving the user's request for transfer, linkage engine 108 transforms a business process model created by modeling tool 104 into a set of business requirements. In one embodiment, linkage engine 108 creates relationships between the business requirements based on the structure of the business process model. In one embodiment, linkage engine 108 updates data stored in business requirements database 110 with additions or changes to a corresponding business process model in modeling tool 104. Alternatively, linkage engine 108 allows the creation of a business process model in modeling tool 104 from data stored in business requirements database 110. For example, the user may modify business requirements in business requirements database

110 directly. Then, the modified business requirements may be exported to an external file that can subsequently be used as an import file 114 into modeling tool 104. Linkage engine 106 provides an interface for importing data from file 114 to modeling tool 104. In one embodiment, file 114 has a specific format such as a spreadsheet format or comma-separated-values (CSV) format.

[0036] In one embodiment, linkage engine 108 provides for export of data stored in business requirements database to reporting database 112. Once the user requests such export (e.g., by selecting a designated button), linkage engine 108 extracts the data from business requirements database 110 and transfers it to reporting database 112.

Preparation for Presentation of Business Process Information

[0037] **Figure 2** is flow diagram of one embodiment of a process 200 for facilitating the presentation of business process information. The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

[0038] Referring to **Figure 2,** process 200 begins with processing logic identifying an application product (processing block 202). In one embodiment, the application product is a standard product defined for a specific industry based on the best practices found in this industry. Alternatively, the application product may not be a standard application product (e.g., it may be a complete or incomplete customer-specific application product being tested).

[0039] Next, processing logic associates a business process model pertaining to the application product with a set of views illustrating the realization of the business process in the application product (processing block 204). In one embodiment, the business process model is created in a modeling language from data contained in an input file (e.g., a file having a spreadsheet format or CSV format). As described above, each view is an image representing a particular user interface within the application product. In one embodiment, each view is assigned an identifier (e.g., a view name), and the association between the business process model and the views is accomplished by linking each view to one or more activities within the business process model.

[0040] **Figure 3** is a flow diagram of one embodiment of a process 300 for associating a business process model with a set of views. The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

[0041] Referring to **Figure 3**, process 300 begins with processing logic identifying a set of user interfaces within an application product (processing block 302). Next, processing logic creates views corresponding to the user interfaces (processing block 304) and stores view identifiers in a repository such as linkage repository 108 (processing block 306). In one embodiment, each view identifier are stored with the application product identifier, an identifier of the corresponding user interface if this identifier is different from the view identifier, and an identifier of a collection of user interfaces that includes the user interface associated with the specific view. The collection of user interfaces (also referred to as a screen) is formed based on a particular activity that the user interfaces are designated to perform. In one embodiment, the view (i.e., the image) is also stored in the repository with the view identifier. Alternatively, the views are stored in a separate file.

[0042] Further, each view identifier is mapped to one or more activities represented in the business process model. In one embodiment, in which the code or notations within the application product refer to the activities represented in the business process model (e.g., by the activity name or other identifier), the mapping is done based on this code or notations.

[0043] Afterwards, processing logic associates each view with one or more activities in the repository (processing block 308).

[0044] In one embodiment, processing logic creates a file for use during the presentation of business process data to the user. The file may store the name of the application product, the name of each screen (i.e., a collection of user interfaces combined for a specific activity) within the application product, and the name of each view associated with the screen.

Presentation of Business Process Data

[0045] **Figure 4** illustrates a user interface 400 for presenting business process data to a user, according to one embodiment of the present invention. It should be noted that any known in the art modeling tool can be used to present business process data to the user without loss of generality.

[0046] Referring to **Figure 4**, user interface 400 displayed on a portion of a computer screen consists of two parts: a business process part 402 and a screen flow part 404. Part 402 displays a business process model 402, and part 404 displays a set of views 404. The business process model created for a specific application product represents a number of business process steps (referred to herein as activities) 406. Views 408 are images (e.g., bitmap images) representing user interfaces provided by the application product. Views 408 illustrate the realization of the business process in the application product. The display of the business process model together with the corresponding views enhances the visual representation of the functionality provided by the application product, thus bringing a greater understanding of the business process realized in the application product.

[0047] In one embodiment, the user is allowed to modify the set of views displayed with the business process model. Specifically, the user can request to delete a view, add a new view or edit an existing view (e.g., change the image of the view, the name of the view, the size of the view, etc.).

[0048] **Figures 5A – 5C** illustrate exemplary user interfaces facilitating an addition of a new view to a diagram, according to one embodiment of the present invention.

[0049] Referring to **Figure 5A**, a user interface 500 displayed on a portion of a computer screen contains a business process model diagram 502 and a screen flow diagram 504 including a set of views. When the user decides to add a view, the user adds a new object to the screen flow diagram 504 (this object is shown as a "new state" 506) and invokes a submenu (e.g., by clicking on new state 506 using the right button of the mouse). In the submenu, the user selects an "Add a View" option.

[0050] Referring to **Figure 5B**, in response to the user selection of the "Add a View" option, a new window 522 opens up on top of user interface 500. Window 522 displays a form that allows the user to select the application product associated with the view, specify a configuration file 524, the view sizes 526, a metafile, and other information. Configuration file 524 is the file that identifies screens within the product application and corresponding views. The metafile stores bitmap images of the views.

[0051] Once the user specifies the configuration file and selects the product application, a list of screens and views appear in the form 522 as shown in **Figure 5C**. The user can then select the screen and the view, specify the other necessary information, and click on an Apply button. In response, the state object changes into a bitmap image 544, thus adding a new view to the screen flow diagram.

[0052] In one embodiment, the user is provided with an option of requesting to navigate to an actual user interface in the application product. For example, the user may be allowed to click on a view using the right button of the mouse and selecting a specific option (e.g., "Navigate to a View in Application"). In response, an actual user interface is displayed in the execution mode.

[0053] **Figure 6** is a flow diagram of a process 600 for facilitating the navigation to a view in an application product. The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

[0054] Referring to **Figure 6**, process 600 begins with processing logic receiving a user request to navigate to a view in an application product (processing block 302). In response, processing logic determines an identifier of the view (e.g., the view name) (processing block 304). In one embodiment, in which view identifiers differ from identifiers of corresponding user interfaces, processing logic also determines the identifier of the user interface based on data stored in a repository (e.g., linkage repository 108).

[0055] Further, at processing block 606, processing logic passes a command to the application product (which, in one embodiment, is open in the background), specifying the user interface and requesting to display the user interface in the execution mode. The application product receives the command and executes the code associated with the specified user interface. As a result, the user can observe the actual user interface and interact with it (e.g., enter data, select buttons, etc.).

[0056] In one embodiment, the user is allowed to request a transfer of a business process model to a business requirement database. In response, a set of business requirements is created and stored in the business requirements database. In one embodiment, the set of business requirements is stored under different requirement types (e.g., process type, activity type, view type, etc.) and the relationships between business requirements of different types are defined and maintained based on the structure of the business process model.

[0057] In one embodiment, an addition or a change to the business process model is automatically reflected in the business requirements database. In another embodiment, the additions and changes are not individually reflected. Instead, the user has to specifically request the transfer upon finishing the modifications, and a new set of business requirements will be created in accordance with the modified business process model.

[0058] In one embodiment, in which the user inputs data directly into the business requirements database, the user is allowed to request an export of business requirements to an external file. This external file can then be used as input by a modeling tool to create a new business process model.

Facilitating Development of Customer-Specific Application Products

[0059] **Figure 7** is a flow diagram of a process 700 for facilitating the development of a customer-specific application product. The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both.

[0060] Referring to **Figure 7**, process 700 begins with processing logic displaying a standard business process model with a set of views (processing block 302). The standard business process model pertains to a standard application product developed for a specific industry based on the best practices found in this industry. The display of the standard business process model together with the set of views enhances the customer's understanding of the functionality provided by the standard application product, assists in the mapping of the customer's business requirements to standard functions and maximizes the use of standard functionality in the customized product.

[0061] Next, processing logic receives a user request to transfer the standard business process model to a business requirements database (processing block 704). In response, processing logic creates a set of standard business requirements and stores it in the business requirements database (processing logic 706).

[0062] Further, the customer may analyze the functionality provided by the standard application product based on the standard business process data presented as described above and proceed with modifications to the standard business process model according to the desired functionality (e.g., modifying activities and views).

[0063] Subsequently, at processing block 708, processing logic receives a user request to transfer the modified business process model to a business requirements database. In response, processing logic creates a set of customer-specific business requirements and stores it in the business requirements database (processing logic 710). As a result, the two sets of business requirements can be compared and analyzed to determine which components of the standard application product need to be modified to provide the functionality desired by the customer.

[0064] In one embodiment, the user is allowed to request a transfer of data from the business requirements database to a reporting database that is used to create various types of reports. These reports can then be used for more efficient analysis of the two sets of business requirements.

An Exemplary Computer System

[0065] **Figure 8** is a block diagram of an exemplary computer system 800 that may be used to perform one or more of the operations described herein. In alternative embodiments, the machine may comprise a network router, a network switch, a network bridge, Personal Digital Assistant (PDA), a cellular telephone, a web appliance or any machine capable of executing a sequence of instructions that specify actions to be taken by that machine.

[0066] The computer system 800 includes a processor 802, a main memory 804 and a static memory 806, which communicate with each other via a bus 808. The computer system 800 may further include a video display unit 810 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)). The computer system 800 also includes an alpha-numeric input device 812 (e.g., a keyboard), a cursor control device 814 (e.g., a mouse), a disk drive unit 816, a signal generation device 820 (e.g., a speaker) and a network interface device 822.

[0067] The disk drive unit 816 includes a computer-readable medium 824 on which is stored a set of instructions (i.e., software) 826 embodying any one, or all, of the methodologies described above. The software 826 is also shown to reside, completely or at least partially, within the main memory 804 and/or within the processor 802. The software 826 may further be transmitted or received via the network interface device 822. For the purposes of this specification, the term " computer-readable medium" shall be taken to include any medium that is capable of storing or encoding a sequence of instructions for execution by the computer and that cause the computer to perform any one of the methodologies of the present invention. The term "computer-readable medium" shall accordingly be taken to included, but not be limited to, solid-state memories, optical and magnetic disks, and carrier wave signals.

[0068] Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims which in themselves recite only those features regarded as essential to the invention.